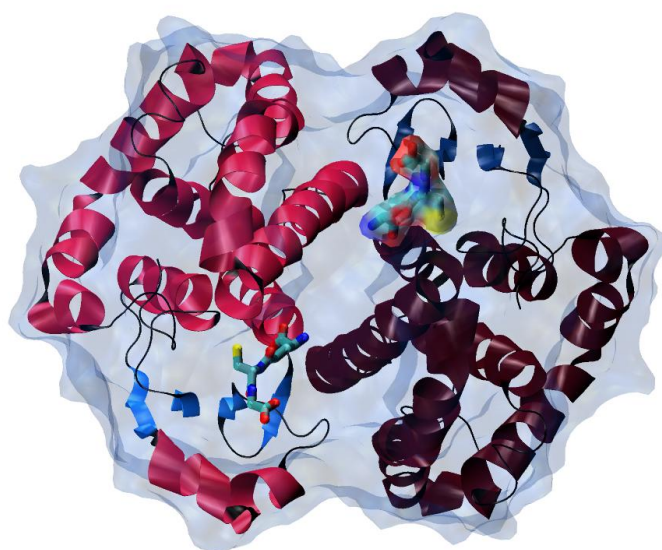


University of Illinois at Urbana-Champaign  
Beckman Institute for Advanced Science and Technology  
Theoretical and Computational Biophysics Group

# TOPOLOGY FILE TUTORIAL

---



Aaron Oakley

Timothy Isgro

Yi Wang

Current Editors:

Christopher Mayne

Po-Chao Wen

February 2012

Please visit [www.ks.uiuc.edu/Training/Tutorials/](http://www.ks.uiuc.edu/Training/Tutorials/) to get the latest version of this tutorial, to obtain more tutorials like this one, or to join the **tutorial-l@ks.uiuc.edu** mailing list for additional help.

## Contents

<b>1</b>	<b>Examining the Topology File</b>	<b>7</b>
1.1	L- and D- Amino Acids . . . . .	8
1.2	Lysine and Ornithine . . . . .	13
1.3	4-Hydroxyproline . . . . .	19
<b>2</b>	<b>Glutathione</b>	<b>21</b>
2.1	Creating the Topology File . . . . .	21
2.2	Using the Topology File . . . . .	28
2.3	Solvating and Ionizing . . . . .	32
<b>3</b>	<b>Solution to Norleucine Problem</b>	<b>34</b>

## Introduction

Often, one encounters the need in molecular dynamics to simulate molecules for which topology and parameter information does not exist. In many cases, parameter development is necessary, but in others it may not be. This tutorial introduces how to create this information based on existing topology information for other molecules, without the need for new parameter development.

The tutorial is subdivided in three sections. The first one introduces the method of topology file creation and situations when it is appropriate and inappropriate. The second section utilizes the method to create a topology file for a tripeptide bound to an enzyme and performs a simulation of the system. The last section provides the solution to a topology file creation problem posed in the tutorial.

Throughout the text, some material will be presented in separate “boxes”. Some of these boxes include complementary information to the tutorial, such as information about the biology of the molecules presented, and tips or warnings one should be aware of. These boxes are not required for understanding the tutorial and may be skipped if you are short on time, but this is not recommended.

Boxes with an exclamation sign are especially important and should not be skipped.

If you have any questions or comments on this tutorial, please email the TCB Tutorial mailing list at [tutorial-l@ks.uiuc.edu](mailto:tutorial-l@ks.uiuc.edu). The mailing list is archived at [http://www.ks.uiuc.edu/Training/Tutorials/mailling\\_list/tutorial-l/](http://www.ks.uiuc.edu/Training/Tutorials/mailling_list/tutorial-l/).

## Required Programs

The following programs are required for this tutorial:

- **VMD:** The tutorial assumes that you already have a working knowledge of VMD, which is available at <http://www.ks.uiuc.edu/Research/vmd/> (for all platforms)
  - The VMD tutorial is available at <http://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/>
- **NAMD:** The latter sections assume that NAMD has been correctly installed on your computer. However, NAMD is not needed to complete this tutorial, but it is recommended. For installation instructions, please refer to the NAMD Users's Guide.
  - The NAMD tutorial is available in both Unix/MacOSX and Windows versions:  
<http://www.ks.uiuc.edu/Training/Tutorials/namd/namd-tutorial-unix-html/>  
<http://www.ks.uiuc.edu/Training/Tutorials/namd/namd-tutorial-win-html/>
- a text editor of your choice (we offer a few easy-to-use recommendations):
  - UNIX: nedit ([www.nedit.org](http://www.nedit.org))
  - Windows XP: WordPad (included with OS). We recommend using WordPad as opposed to NotePad for this tutorial. However, please ensure that you save any files in this tutorial as **.txt** files, as opposed to **.rtf** or **.doc** files.
  - Mac OS X: Smultron ([smultron.sourceforge.net](http://smultron.sourceforge.net)), TextEdit (included with OS)
- a command prompt, such as a terminal in UNIX, Terminal.app in Mac OS X, or the DOS command prompt in Windows.

## Getting Started

If you downloaded the tutorial from the web, the files that you will be needing can be found in a directory called `topology-tutorial-files`. If you received the files through a workshop, they can be found in the same directory under the path `~/Workshop/topology-tutorial`.

- **Unix/Mac OS X Users:** In a Terminal window type:

```
cd <path to topology-tutorial-files directory>
```

You can list the content of this directory, by using the command `ls`.

- **Windows Users:** Navigate to the `topology-tutorial-files` directory using Windows Explorer.

This will place you in the directory containing all the necessary files. In the figure below, you can see the structure of this directory.

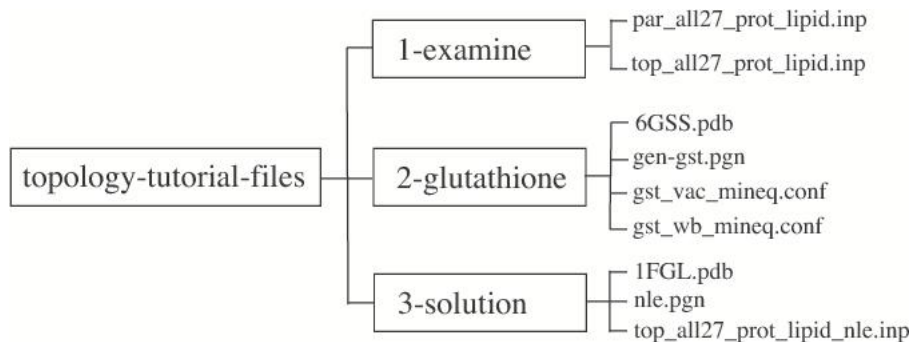


Figure 1: Directory Structure for tutorial exercises. Output for all simulations is provided in an “example-output” subdirectory within each folder shown.

*In this section you will learn about typical situations in which you may need to create your own topology file for a given protein or biological molecule. You will learn about biological systems which utilize amino acids other than the “naturally occurring” 20 and how you may simply be able to alter topology file entries to account for non-standard amino acids without needing to do any new parameterization.*

## 1 Examining the Topology File

Consider the diatomic hydrogen molecule,  $H_2$ . In molecular dynamics simulations, this molecule is modeled as two point masses connected by a simple spring. In this respect, two sets of information are needed to simulate the  $H_2$  molecule:

- atomic connectivity information (to tell the MD program that one atom is bonded to the other)
- the spring stiffness and equilibrium (bond) length

The above simple example demonstrates that if you wish to simulate a more complicated molecule, the MD force field will come in two parts:

- topology file (`top_all127_prot_lipid.inp`)
- parameter file (`par_all127_prot_lipid.inp`)

The topology file defines which atoms are connected to one another through chemical bonds. Different atomic topologies are shown in Figure 2. The topology file may specify “bonds” (2 atoms connected), “angles” (3 atoms connected), and “dihedrals” (4 atoms connected linearly). If you have a complete list of 2-atom bonds, the angle and dihedral information may be inferred from it. Thus, angle and dihedral information is extraneous and may be omitted. Given a complete atom list and set of bonds, psfgen is able to construct the correct topology for the molecule. However, “improper” angles, which are used to restrain chiral and planar centers must be specified explicitly. The improper interaction is used to keep the amino acid peptide bond planar.

The parameter file quantifies the variables which are used in the force field potential energy. It gives parameters such as the stiffness and equilibrium value of an angle between 3 atoms, etc.

Today, we are going to use the topology and parameter files listed above: `top_all127_prot_lipid.inp` and `par_all127_prot_lipid.inp`. These files contain the CHARMM22 protein and the CHARMM27 lipid topologies and parameters.

The CHARMM force field contains topology and parameter information for the standard 20 amino acids, lipids, nucleic acids, and some other organic molecules. Thus, one can simulate any protein, DNA, or molecular systems which are composed of these basic “building blocks”. But what if you wanted

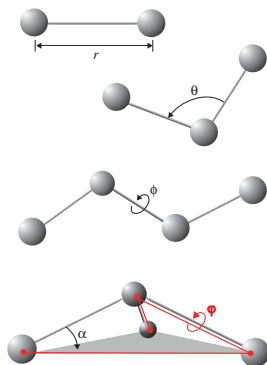


Figure 2: Atomic topologies. From top to bottom, bond, angle, dihedral, and improper topologies are shown. Potential energies are calculated by measuring bond distance ( $r$ ), angle ( $\theta$ ), dihedral ( $\phi$ ), and improper angle ( $\varphi$ ). Note that the improper planarity deviation ( $\alpha$ ) is not used in the potential energy model, but is shown to clarify that improper interactions are used to establish molecular planarity.

to simulate a system that contained a molecule not described in the CHARMM topology and parameter files? At first glance, it might seem like you need to build an entire set of new parameters for your molecule, a process which may involve computationally expensive quantum chemistry calculations. However, this might not be the case.

There are a wide variety of natural and non-natural amino acids and other molecules that can be simulated by piecing together parameters already in the CHARMM parameter set. **You are restricted only by the bond, angle, improper and dihedral angles already available for your atom types.** If you wish to study a protein system involving non-natural amino acids, it may be possible to run simulations using existing parameters for similar standard amino acids. In this manner, you will be saved much effort in not being forced to develop new parameters for your system.

## 1.1 L- and D- Amino Acids

Imagine you have a peptide composed of both L- and D-amino acids. Cyclosporin, an immunosuppressant drug, for example, contains D-alanine. D-alanine is also used in bacterial cell wall synthesis. There are currently 46 files in the Protein Data Bank which contain D-alanine (Example, 1ICO). You might wish to simulate the molecule, but the standard CHARMM22 force field contains only L-amino acids. However, since the molecules are identical in all respects



except for their chirality, the parameters for D-amino acids are identical to their L-configured counterparts. Figure 3 shows the chiral difference between both molecules.

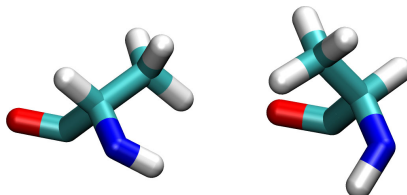


Figure 3: L-alanine (left side) and D-alanine (right-side). Note the difference in chirality around the  $\alpha$  carbon atom. (The different backbone geometries are irrelevant.)

We will develop the new topology file for D-alanine using the existing topology file for L-alanine.

- **Unix/Mac OS X Users:** In a terminal window go to your **1-examine** directory and open the topology file **top\_all127\_prot\_lipid.inp** with your text editor:

```
cd 1-examine
nedit top_all127_prot_lipid.inp
```

- **Windows Users:** Navigate to your **1-examine** directory using Windows Explorer and open the topology file **top\_all127\_prot\_lipid.inp** with your text editor by right-clicking the file, selecting the **Open With...** option and choosing the program WordPad.

- 1 Find the topology entry for alanine using the **Search** or **Find** feature in your text editor by typing "**ala**" in the search field.
- 2 The entry for alanine is shown on the next page.

# Alanine Topology Entry

```

RESI ALA          0.00
GROUP
ATOM N   NH1    -0.47 !   |
ATOM HN  H       0.31 !   HN-N
ATOM CA  CT1     0.07 !   |   HB1
ATOM HA  HB      0.09 !   |   /
GROUP      !   HA-CA--CB-HB2
ATOM CB  CT3    -0.27 !   |   \
ATOM HB1 HA      0.09 !   |   HB3
ATOM HB2 HA      0.09 !   O=C
ATOM HB3 HA      0.09 !   |
GROUP      !
ATOM C    C       0.51
ATOM O    O      -0.51
BOND CB CA N HN N CA
BOND C CA C +N CA HA CB HB1 CB HB2 CB HB3
DOUBLE O C
IMPR N -C CA HN C CA +N O
DONOR HN N
ACCEPTOR O C
IC -C CA *N HN 1.3551 126.4900 180.0000 115.4200 0.9996
IC -C N CA C 1.3551 126.4900 180.0000 114.4400 1.5390
IC N CA C +N 1.4592 114.4400 180.0000 116.8400 1.3558
IC +N CA *C O 1.3558 116.8400 180.0000 122.5200 1.2297
IC CA C +N +CA 1.5390 116.8400 180.0000 126.7700 1.4613
IC N C *CA CB 1.4592 114.4400 123.2300 111.0900 1.5461
IC N C *CA HA 1.4592 114.4400 -120.4500 106.3900 1.0840
IC C CA CB HB1 1.5390 111.0900 177.2500 109.6000 1.1109
IC HB1 CA *CB HB2 1.1109 109.6000 119.1300 111.0500 1.1119
IC HB1 CA *CB HB3 1.1109 109.6000 -119.5800 111.6100 1.1114

```

The entries in the topology file provide the following information:

## 3 RESI ALA 0.00

indicates a new residue with the name **ALA** and total charge of **0.00**.

## 4 GROUP

indicates that the following atoms (up to the next **GROUP** command) are part of a “group” of atoms which carries an integer charge. Groups are often made up of atoms which interact electronically, sharing electron density.

## 5 ATOM N NH1 -0.47 ! |

indicates a new atom with the name **N**, the type **NH1**, and a charge of **-0.47**. The **!** sign is a comment signal. Anything appearing after it on the line will be ignored. Thus, the stick representation of alanine is simply a comment in the file. It is only there so the user may see the topology of the residue. Note that atoms **N**, **HN**, **CA**, and **HA** form a group with net charge **0.00**.

## 6 BOND CB CA N HN N CA

indicates sets of 2 atoms which are connected by a single bond. The bonds are created between consecutive atoms. Thus, one bond is placed between

atoms **CB** and **CA**, one is placed between **N** and **HN**, and one between **N** and **CA**. The **BOND** entries on the next line are read exactly the same way, and are put on a new line simply for readability.

#### 7 DOUBLE O C

simply indicates that the atoms named **O** and **C** are bonded. **DOUBLE** is a synonym for **BOND** and will not affect the resulting psf file. The fact that there is a double bond between the atoms is accounted for in the parameters, not in the topology.

#### 8 IMPR N -C CA HN C CA +N O

indicates sets of 4 atoms for which the improper bond interaction will be calculated. The chiral center is listed first. An improper bond is created between atoms **N**, **C**, **CA**, and **HN**, with **N** being at the center, and one is created between **C**, **CA**, **N**, and **O** with **C** at the center. Recall, improper bonds are created to maintain planarity, and these two maintain the planarity of the protein backbone.

#### 9 DONOR and ACCEPTOR lines

are not used by the CHARMM force field or NAMD. They indicate which atoms may act as hydrogen bond donors and acceptors, but since the CHARMM force field no longer includes an explicit hydrogen bonding energy term, these entries are ignored. Note that any hydrogen bonds you observe in your simulations using the latest CHARMM force field are purely the result of electrostatic interactions.

#### 10 IC -C CA \*N HN 1.3551 126.4900 180.0000 115.4200 0.9996

**IC** stands for "internal coordinates". They provide a complete set of coordinates for atoms *relative* to other atoms in the residue. Thus, if atoms are missing from a structure (such as H atoms from an x-ray crystallography PDB file), the complete molecule may still be built based on the positions of existing atoms. The **IC** records define the bond length, angle, dihedral (or improper) angle for groups of 4 atoms. The pattern may come in 2 forms:

- **IC A B C D** [bond(AB)] [angle(ABC)] [dihedral(ABCD)] [angle(BCD)] [bond(CD)]
  - **IC A B \*C D** [bond(AC)] [angle(BCA)] [improper(ABCD)] [angle(BCD)] [bond(CD)].
- The **\*** next to the **C** atom indicates that it is at the center of an improper angle definition. No **\*** indicates that the 4 atoms do not have an improper topology.

In the example above, atoms **C** and **N** have a bond length of 1.3551 Å, atoms **CA**, **N**, and **C** have an angle of 126.4900° between them, atoms **C**, **CA**, **N**, and **HN** have a 180.0000° improper angle, atoms **CA**, **N**, and **HN** have an angle of 115.4200° between them, and atoms **N** and **HN** have a bond length of 0.9996 Å. Atom **N** is the central atom in the improper topology.



**Internal Coordinates.** Specifying IC's is not necessary if you already have all the atoms in your PDB file. The entries are simply there to have a way to create missing atoms from the positions of present ones.

Now, how can we use the above topology entry for L-alanine to create a topology for D-alanine?

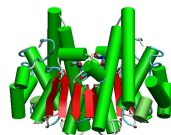
- 11 Go to your text editor and copy the existing entry for alanine. Use the mouse to highlight every alanine entry line (from `RESI ALA 0.00` to `IC HB1 CA *CB HB3 1.1109 109.6000 -119.5800 111.6100 1.1114`). Then click `Edit → Copy`.
- 12 Paste the copied entry below the original alanine entry by using the `Edit → Paste` option.
- 13 Edit your copied alanine entry to make it D-alanine by changing the following lines:

```
Change:  RESI ALA 0.00
to:      RESI DAL 0.00
Change:  IC N C *CA CB 1.4592 114.4400 123.2300 111.0900 1.5461
to:      IC N C *CA CB 1.4592 114.4400 -123.2300 111.0900 1.5461
Change:  IC N C *CA HA 1.4592 114.4400 -120.4500 106.3900 1.0840
to:      IC N C *CA HA 1.4592 114.4400 120.4500 106.3900 1.0840
```

The only difference between the two molecules is the inverse chirality of the methyl side chain and HA atom about the CA, or  $\alpha$ -carbon, atom. Thus, by changing 123.23 to -123.23 in the first IC, we change the L-chiral center at the CA atom to a D- chiral centre. The same principle applies to the second IC command.

- 14 Click `File → Save As...`, enter `top_all127_prot_lipid_d-ala.inp` as your new filename, and click `Enter`. You have now created a topology file with D-alanine included!

**Windows Users:** Make sure you save the file in `.txt` format.

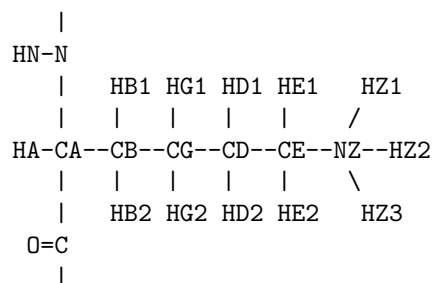


**L- and D- Amino Acids in Biology.** Peptidoglycan is an exclusively bacterial polymer. Many antibiotics interfere with its synthesis and its components. Peptidoglycan is unique among biological polymers because it contains both L- and D-isomers of its constituent amino acids. The D-alanine found in some peptidoglycans is produced by the racemization of the standard amino acid L-alanine by alanine racemase.

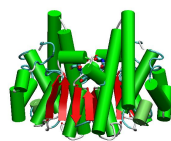
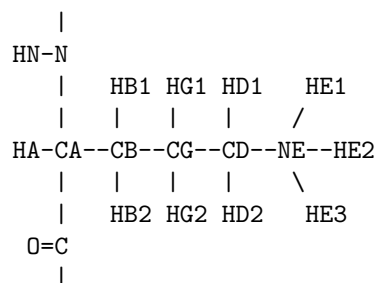
## 1.2 Lysine and Ornithine

Next, let's consider a natural amino acid that is not normally incorporated into proteins. Ornithine is similar to lysine but is one carbon atom shorter in its side-chain:

Lysine:



Ornithine:



**Ornithine.** Ornithine is not one of the 20 standard amino acids, but the molecule has a great deal of relevance in biological systems. It is one of the products of the action of the enzyme arginase on L-arginine, creating urea. Therefore, ornithine is a central part of the urea cycle, which allows for the disposal of excess nitrogen. Ornithine is not an amino acid coded for by DNA, and in that sense, is not involved in protein synthesis. However, in mammalian non-hepatic tissues, the main use of the urea cycle is in arginine biosynthesis, so as an intermediate in metabolic processes, it is quite important. Ornithine is also the starting point for the synthesis of polyamines such as putrescine, via the action of ornithine decarboxylase (E.C. 4.1.1.17). Currently, 17 PDB structures contain ornithine, one example of which is 1B2H, which contains a tripeptide of Lys-Orn-Lys.

Let's say that you want to probe the effect of side-chain length of a particular lysine residue within your protein. You could run a simulation incorporating ornithine into the protein. However, a topology entry would be needed for the residue. Furthermore, if existing atom types were suitable to be used in the

ornithine molecule, you would not need to create parameters. In this section, we will demonstrate how to create a topology entry for ornithine.

Construction of a topology entry for ornithine (or any new molecule) from existing topology entries involves assigning atom types and charges based on similar molecules. Therefore, it is useful to sketch out the new molecule in order to determine which parts are similar to other molecules for which topology and parameter information already exists. Since ornithine is very similar to lysine, we have sketched lysine below and isolated its groups. Then, we have matched up similar groups in ornithine.

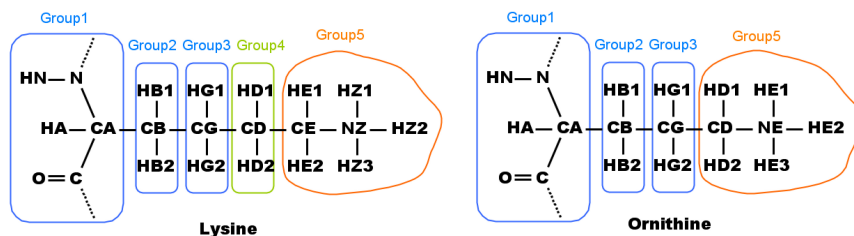


Figure 4: Diagram comparing the topologies of lysine (left) and ornithine (right). Ornithine is very similar to lysine, except for an aliphatic carbon along its side chain (Group 4).

As you can see, the above case is rather simple. An ornithine topology entry may be created from lysine by simply removing one of lysine's aliphatic carbon atoms, along with the hydrogen atoms bonded to it. Diagrams such as the one above are very useful when constructing a topology entry.

Now that we know which lysine groups will be used to construct the ornithine topology entry, what exact changes are needed to make it? We will begin by examining the lysine topology entry

- 1 In your `1-examine` directory, open your D-alanine topology file with your text editor by:
  - **Unix/Mac OS X Users:** typing the following in your terminal window:  

```
nedit top_all127_prot_lipid_d-ala.inp
```
  - **Windows Users:** right-clicking the file `top_all127_prot_lipid_d-ala.inp`, selecting the `Open With...` option and choosing the program Word-Pad.
- 2 Find the topology entry for lysine using the **Search** or **Find** feature in your text editor by typing "lys" in the search field.
- 3 The entry for lysine is shown on the next page.

Lysine Topology Entry

```

RESI LYS          1.00
GROUP
ATOM N      NH1   -0.47 !      |
ATOM HN     H      0.31 !    HN-N
ATOM CA     CT1   0.07 !      |   HB1 HG1 HD1 HE1   HZ1
ATOM HA     HB     0.09 !      |   |   |   |   /
GROUP      !   HA-CA--CB--CG--CD--CE--NZ--HZ2
ATOM CB     CT2   -0.18 !      |   |   |   |   \
ATOM HB1    HA     0.09 !      |   HB2 HG2 HD2 HE2   HZ3
ATOM HB2    HA     0.09 !      |   O=C
GROUP      !      |
ATOM CG     CT2   -0.18
ATOM HG1    HA     0.09
ATOM HG2    HA     0.09
GROUP
ATOM CD     CT2   -0.18
ATOM HD1    HA     0.09
ATOM HD2    HA     0.09
GROUP
ATOM CE     CT2   0.21
ATOM HE1    HA     0.05
ATOM HE2    HA     0.05
ATOM NZ     NH3   -0.30
ATOM HZ1    HC     0.33
ATOM HZ2    HC     0.33
ATOM HZ3    HC     0.33
GROUP
ATOM C      C      0.51
ATOM O      O     -0.51
BOND CB CA   CG CB   CD CG   CE CD   NZ CE
BOND N HN    N CA   C CA
BOND C +N    CA HA   CB HB1  CB HB2  CG HG1
BOND CG HG2  CD HD1  CD HD2  CE HE1  CE HE2
DOUBLE O C
BOND NZ HZ1  NZ HZ2  NZ HZ3
IMPR N -C CA HN  C CA +N O
DONOR HN N
DONOR HZ1 NZ
DONOR HZ2 NZ
DONOR HZ3 NZ
ACCEPTOR O C
IC -C  CA  *N  HN   1.3482 123.5700 180.0000 115.1100 0.9988
IC -C  N   C   C    1.3482 123.5700 180.0000 107.2900 1.5187
IC N   CA  C   +N   1.4504 107.2900 180.0000 117.2700 1.3478
IC +N  CA  *C   O    1.3478 117.2700 180.0000 120.7900 1.2277
IC CA  C   +N  +CA   1.5187 117.2700 180.0000 124.9100 1.4487
IC N   C   *CA  CB   1.4504 107.2900 122.2300 111.3600 1.5568
IC N   C   *CA  HA   1.4504 107.2900 -116.8800 107.3600 1.0833
IC N   CA  CB   CG   1.4504 111.4700 180.0000 115.7600 1.5435
IC CG  CA  *CB  HB1   1.5435 115.7600 120.9000 107.1100 1.1146
IC CG  CA  *CB  HB2   1.5435 115.7600 -124.4800 108.9900 1.1131
IC CA  CB   CG   CD   1.5568 115.7600 180.0000 113.2800 1.5397
IC CD  CB   *CG  HG1   1.5397 113.2800 120.7400 109.1000 1.1138
IC CD  CB   *CG  HG2   1.5397 113.2800 -122.3400 108.9900 1.1143
IC CB  CG   CD   CE   1.5435 113.2800 180.0000 112.3300 1.5350
IC CE  CG   *CD  HD1   1.5350 112.3300 122.2500 108.4100 1.1141
IC CE  CG   *CD  HD2   1.5350 112.3300 -121.5900 108.1300 1.1146
IC CG  CD   CE   NZ   1.5397 112.3300 180.0000 110.4600 1.4604
IC NZ  CD   *CE  HE1   1.4604 110.4600 119.9100 110.5100 1.1128
IC NZ  CD   *CE  HE2   1.4604 110.4600 -120.0200 110.5700 1.1123
IC CD  CE   NZ   HZ1   1.5350 110.4600 179.9200 110.0200 1.0404
IC HZ1 CE   *NZ  HZ2   1.0404 110.0200 120.2700 109.5000 1.0402
IC HZ1 CE   *NZ  HZ3   1.0404 110.0200 -120.1300 109.4000 1.0401

```

- 4 Copy the existing lysine entry. Use the mouse to highlight every lysine entry line (from RESI LYS 1.00 to IC HZ1 CE \*NZ HZ3 1.0404 110.0200 -120.1300 109.4000 1.0401). Then click Edit → Copy.
- 5 Paste the copied entry below the original lysine entry by using the Edit → Paste option.

You will edit your copied lysine entry to make it an ornithine entry by making the following changes:

- 6 Change: RESI LYS 1.00  
to: RESI ORN 1.00
- 7 Delete the appropriate carbon atom and hydrogen atoms. We will delete the  $\epsilon$ -carbon (CE) atom and the hydrogens bonded to it (HE1 and HE2), but we must be careful! In deleting the  $\epsilon$ -carbon, we are deleting atoms which are part of a group with NZ. Thus, the  $\delta$ -carbon and its hydrogens will become part of this group.
  - Delete the following lines:
 

```
GROUP
ATOM CD CT2 -0.18
ATOM HD1 HA 0.09
ATOM HD2 HA 0.09
```
  - Change the following lines:
 

```
Change: ATOM CE CT2 0.21
to:      ATOM CD CT2 0.21
Change: ATOM HE1 HA 0.05
to:      ATOM HD1 HA 0.05
Change: ATOM HE2 HA 0.05
to:      ATOM HD2 HA 0.05
Change: ATOM NZ NH3 -0.30
to:      ATOM NE NH3 -0.30
Change: ATOM HZ1 HC 0.33
to:      ATOM HE1 HC 0.33
Change: ATOM HZ2 HC 0.33
to:      ATOM HE2 HC 0.33
Change: ATOM HZ3 HC 0.33
to:      ATOM HE3 HC 0.33
```
- 8 In considering bonding, we must delete the bonds between CE and all other atoms, and take care to rename the NZ and HZ atoms appropriately.
 

```
Change: BOND CB CA CG CB CD CG CE CD NZ CE
to:      BOND CB CA CG CB CD CG NE CD
Change: BOND CG HG2 CD HD1 CD HD2 CE HE1 CE HE2
to:      BOND CG HG2 CD HD1 CD HD2
Change: BOND NZ HZ1 NZ HZ2 NZ HZ3
to:      BOND NE HE1 NE HE2 NE HE3
```



- 9 Even though the hydrogen bond `DONOR` terms are unused, we still change them to maintain consistency in atom names.

```
Change:  DONOR HZ1 NZ
to:      DONOR HE1 NE
Change:  DONOR HZ2 NZ
to:      DONOR HE2 NE
Change:  DONOR HZ3 NZ
to:      DONOR HE3 NE
```

- 10 Since we delete 3 atoms from the lysine molecule in order to create ornithine, 3 IC commands become obsolete.

Delete the following lines:

```
IC CB CG CD CE 1.5435 113.2800 180.0000 112.3300 1.5350
IC CE CG *CD HD1 1.5350 112.3300 122.2500 108.4100 1.1141
IC CE CG *CD HD2 1.5350 112.3300 -121.5900 108.1300 1.1146
```

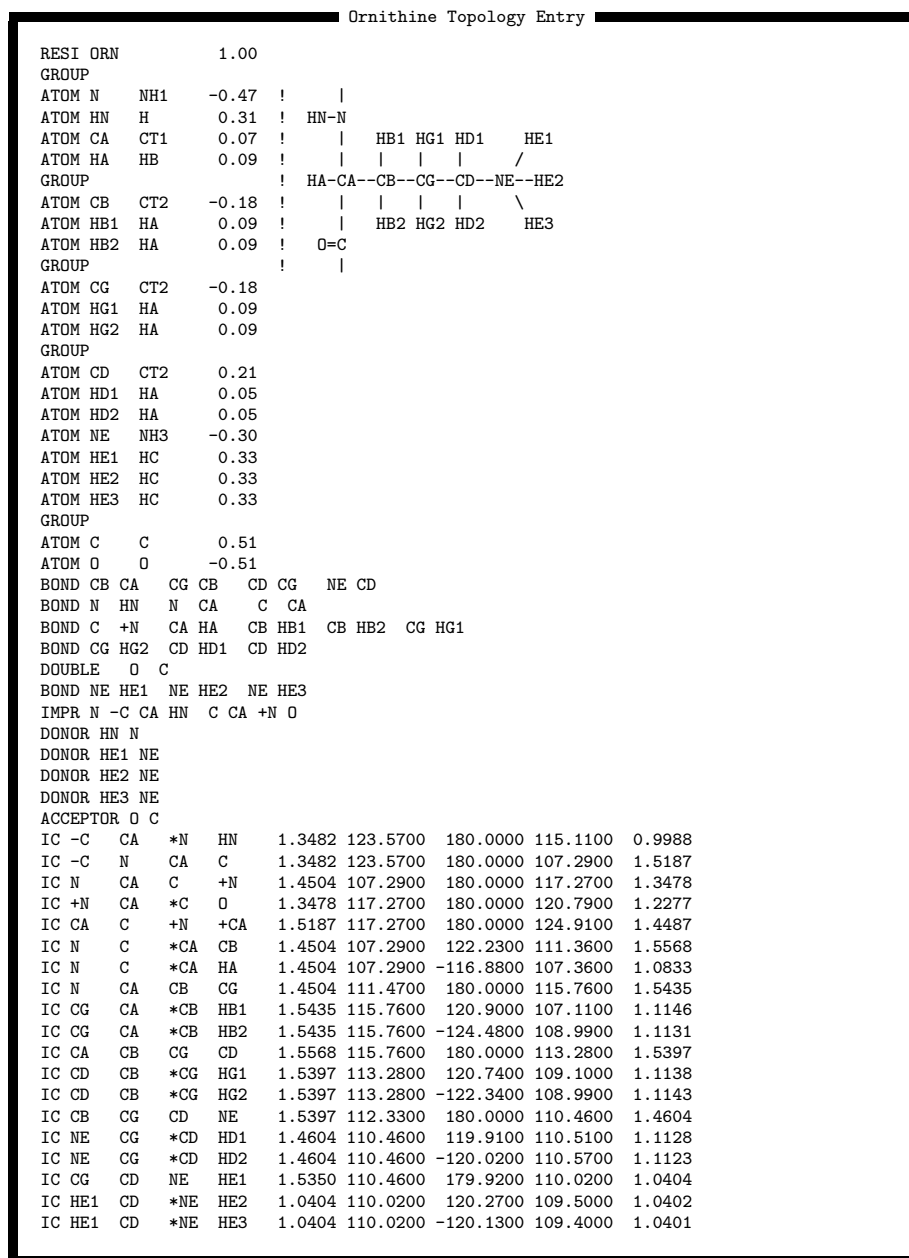
- 11 You need to be very careful when renaming atoms and updating the ICs accordingly. For our ornithine molecule change the atom names in the following lines. The atomic arrangements (bond lengths, angles, etc.) stay the same, since we simply removed one carbon atom from the side chain.

```
Change:  IC CG CD CE NZ 1.5397 112.3300 180.0000 110.4600 1.4604
to:      IC CB CG CD NE 1.5397 112.3300 180.0000 110.4600 1.4604
Change:  IC NZ CD *CE HE1 1.4604 110.4600 119.9100 110.5100 1.1128
to:      IC NE CG *CD HD1 1.4604 110.4600 119.9100 110.5100 1.1128
Change:  IC NZ CD *CE HE2 1.4604 110.4600 -120.0200 110.5700 1.1123
to:      IC NE CG *CD HD2 1.4604 110.4600 -120.0200 110.5700 1.1123
Change:  IC CD CE NZ HZ1 1.5350 110.4600 179.9200 110.0200 1.0404
to:      IC CG CD NE HE1 1.5350 110.4600 179.9200 110.0200 1.0404
Change:  IC HZ1 CE *NZ HZ2 1.0404 110.0200 120.2700 109.5000 1.0402
to:      IC HE1 CD *NE HE2 1.0404 110.0200 120.2700 109.5000 1.0402
Change:  IC HZ1 CE *NZ HZ3 1.0404 110.0200 -120.1300 109.4000 1.0401
to:      IC HE1 CD *NE HE3 1.0404 110.0200 -120.1300 109.4000 1.0401
```



**Creating IC commands.** Remember, IC commands are there so that missing atoms may be placed in your structure. You should have as many IC commands as there are atoms in your molecule or residue.

After making all the corrections, your topology definition for ornithine should look as shown on the next page.

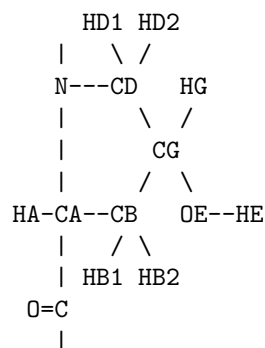


**12** Click File → Save As..., enter `top_all127_prot_lipid_orn.inp` as your new filename, and click Enter. You have now created a topology file with ornithine (and D-alanine) included.

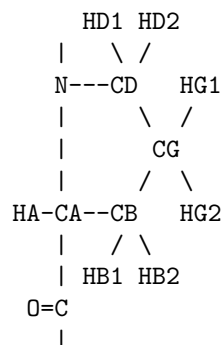
Windows Users: Make sure you save the file in `.txt` format.



4-Hydroxyproline:



Proline:



Although it may not seem to be a very big change in the chemical formula, the replacement of H by OH is not possible to treat with the standard topology library at our disposal.

As you may have already noticed, all the five atoms on the ring of proline have unique atom types. For example, the atom type of the nitrogen atom on proline ring is N which is defined as “proline N”, instead of NH1 which is the normal atom type for peptide nitrogen. The atom with name CG is of type CP2, again, unique to proline. Thus, bonding an O atom to CG would require choosing a type for the O atom, which is impossible, since there is no topology entry with O bonded to a proline or proline-like ring. We simply do not have parameters for oxygen bonded to a 5-member proline ring.

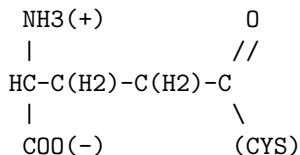
As a result, we would have to re-parameterize this new amino acid if we wished to work with it. New atom types and parameters would be created from quantum chemistry calculations in order to properly model the new ring.

*In this section you will learn how to apply the lessons in Unit 1 from beginning to end, building a new topology file, using it to generate a psf file for the system, and running a new simulation with the topology you created.*

## 2 Glutathione

Now we will try something a bit harder. We are going to build a topology file for an unusual peptide.

Many enzymes utilize a tripeptide called glutathione ( $\gamma$ -Glu-Cys-Gly), or GSH for short. The tripeptide is unusual in that the first amino acid, glutamic acid, is linked to cysteine via its side-chain, rather than the protein backbone:



### 2.1 Creating the Topology File

Once again, by using existing topologies and parameters available within CHARMM22, we will be able to create a topology entry for the  $\gamma$ -Glu residue and run a simulation incorporating the peptide. Since  $\gamma$ -Glu is almost identical to standard glutamic acid, we will use the glutamic acid topology entry as a starting point from which to create out new topology.

- 1 Copy the topology file you created for ornithine to the **2-glutathione** directory and open the file with your text editor:
  - **Unix/Mac OS X Users:** In a terminal window, type:
 

```
cd ../2-glutathione
cp ../1-examine/top_all27_prot_lipid_orn.inp .
nedit top_all27_prot_lipid_orn.inp
```
  - **Windows Users:** Right-click the file **top\_all27\_prot\_lipid\_orn.inp**, select **Copy**, navigate to the **2-glutathione** directory, right-click and select **Paste** to copy the file there. Then right-click the file, select **Open With...**, and choose the program WordPad.
- 2 Find the topology entry for glutamic acid using the **Search** or **Find** feature in your text editor by typing "**resi glu**" in the search field.
- 3 The entry for glutamic acid is shown on the next page.

# Glutamic Acid Topology Entry

```

RESI GLU          -1.00
GROUP
ATOM N    NH1    -0.47 !      |
ATOM HN   H      0.31 !    HN-N
ATOM CA   CT1    0.07 !      |   HB1 HG1   OE1
ATOM HA   HB     0.09 !      |   |   //
GROUP     !    HA-CA--CB--CG--CD
ATOM CB   CT2    -0.18 !      |   |   \
ATOM HB1  HA     0.09 !      |   HB2 HG2   OE2(-)
ATOM HB2  HA     0.09 !      |   O=C
GROUP     !      |
ATOM CG   CT2    -0.28
ATOM HG1  HA     0.09
ATOM HG2  HA     0.09
ATOM CD   CC     0.62
ATOM OE1  OC    -0.76
ATOM OE2  OC    -0.76
GROUP
ATOM C    C      0.51
ATOM O    O     -0.51
BOND CB CA  CG CB CD CG OE2 CD
BOND N HN   N CA C  CA
BOND C +N   CA HA  CB HB1 CB HB2 CG  HG1
BOND CG HG2
DOUBLE O C  CD OE1
IMPR N  -C CA HN C CA +N O
!IMPR OE1 CG OE2 CD
IMPR CD CG OE2 OE1
DONOR HN N
ACCEPTOR OE1 CD
ACCEPTOR OE2 CD
ACCEPTOR O C
IC -C   CA  *N   HN   1.3471 124.4500 180.0000 113.9900 0.9961
IC -C   N   CA   C    1.3471 124.4500 180.0000 107.2700 1.5216
IC N    CA  C    +N   1.4512 107.2700 180.0000 117.2500 1.3501
IC +N   CA  *C   O    1.3501 117.2500 180.0000 121.0700 1.2306
IC CA   C    +N   +CA  1.5216 117.2500 180.0000 124.3000 1.4530
IC N    C    *CA  CB   1.4512 107.2700 121.9000 111.7100 1.5516
IC N    C    *CA  HA   1.4512 107.2700 -118.0600 107.2600 1.0828
IC N    CA  CB   CG   1.4512 111.0400 180.0000 115.6900 1.5557
IC CG   CA  *CB  HB1   1.5557 115.6900 121.2200 108.1600 1.1145
IC CG   CA  *CB  HB2   1.5557 115.6900 -123.6500 109.8100 1.1131
IC CA   CB   CG   CD   1.5516 115.6900 180.0000 115.7300 1.5307
IC CD   CB   *CG  HG1   1.5307 115.7300 117.3800 109.5000 1.1053
IC CD   CB   *CG  HG2   1.5307 115.7300 -121.9600 111.0000 1.1081
IC CB   CG   CD   OE1   1.5557 115.7300 180.0000 114.9900 1.2590
IC OE1  CG   *CD  OE2   1.2590 114.9900 -179.1000 120.0800 1.2532

```

4 Copy the existing glutamic acid entry. Use the mouse to highlight every entry line (from RESI GLU -1.00 to IC OE1 CG \*CD OE2 1.2590 114.9900 -179.1000 120.0800 1.2532). Then click Edit → Copy.

5 Paste the copied entry below the original glutamic acid entry by using the Edit → Paste option.

You will edit your copied glutamic acid entry to make it an  $\gamma$ -glutamic acid entry by making the following changes:

6 Since our  $\gamma$ -Glu residue will no longer have a side chain negative charge, and the “backbone” will have both a positive N-terminus and negative C-terminus, the total charge on the residue is 0.

Change: RESI GLU -1.00  
to: RESI GGL 0.00

- 7 Let's begin by examining the "backbone". Since both  $\gamma$ -Glu "termini" resemble normal N- and C-termini, we should use atom types and charges from the terminal patches in our residue.

Change the first GROUP:

```
GROUP !
ATOM N NH1 -0.47 !
ATOM HN H 0.31 !
ATOM CA CT1 0.07 !
ATOM HA HB 0.09 !
```

to its NTER counterpart:

```
GROUP !
ATOM N NH3 -0.30 !
ATOM HT1 HC 0.33 !
ATOM HT2 HC 0.33 !
ATOM HT3 HC 0.33 !
ATOM CA CT1 0.21 !
ATOM HA HB 0.10 !
```

```

N-terminus Patch Topology Entry
PRES NTER      1.00 ! standard N-terminus
GROUP          ! use in generate statement
ATOM N  NH3    -0.30 !
ATOM HT1 HC     0.33 !           HT1
ATOM HT2 HC     0.33 !           (+)/
ATOM HT3 HC     0.33 ! --CA--N--HT2
ATOM CA  CT1    0.21 ! |  \
ATOM HA  HB     0.10 !  HA   HT3
DELETE ATOM HN
BOND HT1 N HT2 N HT3 N
DONOR HT1 N
DONOR HT2 N
DONOR HT3 N
IC HT1  N  CA  C    0.0000  0.0000  180.0000  0.0000  0.0000
IC HT2  CA  *N  HT1  0.0000  0.0000  120.0000  0.0000  0.0000
IC HT3  CA  *N  HT2  0.0000  0.0000  120.0000  0.0000  0.0000

```

```

C-terminus Patch Topology Entry
PRES CTER      -1.00 ! standard C-terminus
GROUP          ! use in generate statement
ATOM C  CC      0.34 !           OT2(-)
ATOM OT1 OC     -0.67 ! /
ATOM OT2 OC     -0.67 ! -C
DELETE ATOM O    !  \
BOND C  OT2      !   OT1
DOUBLE  C  OT1
!IMPR OT1 CA OT2 C
IMPR C  CA OT2 OT1
ACCEPTOR OT1 C
ACCEPTOR OT2 C
IC N    CA  C    OT2  0.0000  0.0000  180.0000  0.0000  0.0000
IC OT2  CA  *C  OT1  0.0000  0.0000  180.0000  0.0000  0.0000

```

Notice we have left out the ASCII drawing of our new residue. Try to construct it yourself!

- 8 Add the following **GROUP** for the C-terminus taken from the **CTER** patch:

```
GROUP
ATOM C CC 0.34
ATOM OT1 OC -0.67
ATOM OT2 OC -0.67
```

- 9 Now, let's look at the "side chain" through which the peptide bond is formed. In the original glutamic acid residue, the side chain is quite electronegative, and draws electron density from the  $\text{CH}_2$  atoms next to it, so the entire set  $\text{C}_2\text{H}_2\text{O}_2$  is a single **GROUP**. In  $\gamma$ -Glu, however, since the "side chain" more resembles a peptide bond, we separate it into three **GROUPs**:  $\text{CH}_2$ ,  $\text{CH}_2$ , and  $\text{CO}$ .

- Leave the following **GROUP** the way it is:

```
GROUP
ATOM CB CT2 -0.18
ATOM HB1 HA 0.09
ATOM HB2 HA 0.09
```

- Add the following **GROUP**:

```
GROUP
ATOM CG CT2 -0.18
ATOM HG1 HA 0.09
ATOM HG2 HA 0.09
```

- Delete the following **GROUP**:

```
GROUP !
ATOM CG CT2 -0.28
ATOM HG1 HA 0.09
ATOM HG2 HA 0.09
ATOM CD CC 0.62
ATOM OE1 OC -0.76
ATOM OE2 OC -0.76
```

- Alter the following **GROUP** to resemble the C-terminus (taken from **CTER** patch):

```
Change:  ATOM C C 0.51
to:      ATOM CD C 0.51
```

- 10 We also need to account for the bonds which are created and destroyed in changing glutamic acid to  $\gamma$ -Glu, remembering new atom names:



```

Change:  BOND CB CA CG CB CD CG OE2 CD
to:      BOND CB CA CG CB CD CG
Change:  BOND N HN N CA C CA
to:      BOND HT1 N HT2 N HT3 N CA N
Change:  BOND C +N CA HA CB HB1 CB HB2 CG HG1
to:      BOND CD +N CA HA CB HB1 CB HB2 CG HG1 CG HG2 CA C
Change:  BOND CG HG2
to:      BOND C OT2
Change:  DOUBLE O C CD OE1
to:      DOUBLE CD O C OT1

```

- 11** In considering the improper entries, we should examine both the NTER and CTER patches for our “backbone” and the GLU entry for the modified “side chain”. We find a planar structure near both double-bonded oxygens.

```

Change:  IMPR N -C CA HN C CA +N O
to:      IMPR CD CG +N O
Change:  IMPR CD CG OE2 OE1
to:      IMPR C CA OT1 OT2

```

You may delete the IMPR command for glutamic acid which is commented out.

Note the use of – and + signs in the topology entry. They indicate the peptide bonds to the preceding and next amino acids in the sequence, respectively.

- 12** Change the DONOR and ACCEPTOR commands to maintain consistency in atom names.

```

Change:  DONOR HN N
to:      DONOR HT1 N
          DONOR HT2 N
          DONOR HT3 N
Change:  ACCEPTOR OE1 CD
to:      ACCEPTOR O CD
Change:  ACCEPTOR OE2 CD
to:      ACCEPTOR OT1 C
Change:  ACCEPTOR O C
to:      ACCEPTOR OT2 C

```

- 13** The IC commands are the most complicated (as they will always be). We can construct them through a combination of those in patches NTER and CTER and residue GLU.

- Delete the following IC entries:

```

IC -C CA *N HN 1.3471 124.4500 180.0000 113.9900 0.9961
IC -C N CA C 1.3471 124.4500 180.0000 107.2700 1.5216
IC N CA C +N 1.4512 107.2700 180.0000 117.2500 1.3501
IC +N CA *C O 1.3501 117.2500 180.0000 121.0700 1.2306

```

```
IC CA C +N +CA 1.5216 117.2500 180.0000 124.3000 1.4530
IC OE1 CG *CD OE2 1.2590 114.9900 -179.1000 120.0800 1.2532
```

- Leave the following IC entries based on residue GLU:

```
IC N C *CA CB 1.4512 107.2700 121.9000 111.7100 1.5516
IC N C *CA HA 1.4512 107.2700 -118.0600 107.2600 1.0828
IC N CA CB CG 1.4512 111.0400 180.0000 115.6900 1.5557
IC CG CA *CB HB1 1.5557 115.6900 121.2200 108.1600 1.1145
IC CG CA *CB HB2 1.5557 115.6900 -123.6500 109.8100 1.1131
IC CA CB CG CD 1.5516 115.6900 180.0000 115.7300 1.5307
IC CD CB *CG HG1 1.5307 115.7300 117.3800 109.5000 1.1053
IC CD CB *CG HG2 1.5307 115.7300 -121.9600 111.0000 1.1081
```

- Change the following IC entry based on residue GLU:

```
Change: IC CB CG CD OE1 1.5557 115.7300 180.0000 114.9900 1.2590
to:      IC CB CG CD O 1.5557 115.7300 180.0000 114.9900 1.2590
```

- Add the following IC entries based on the NTER and CTER patches:

```
IC HT1 N CA C 0.0000 0.0000 180.0000 0.0000 0.0000
IC HT2 CA *N HT1 0.0000 0.0000 120.0000 0.0000 0.0000
IC HT3 CA *N HT2 0.0000 0.0000 120.0000 0.0000 0.0000
IC N CA C OT2 0.0000 0.0000 180.0000 0.0000 0.0000
IC OT2 CA *C OT1 0.0000 0.0000 180.0000 0.0000 0.0000
```

After making the changes, your topology definition for  $\gamma$ -Glu should look as shown on the next page. We have added comments to keep track of where certain commands came from, and also drawn the correct topology for the molecule in ASCII.

```

γ-Glutamic Acid Topology Entry

RESI GGL      0.00
GROUP        !      HT2
ATOM N      NH3 -0.30 !      |(+ )
ATOM HT1    HC  0.33 !      |
ATOM HT2    HC  0.33 ! HT1-N-HT3 <-- from NTER
ATOM HT3    HC  0.33 !      |
ATOM CA      CT1 0.21 !      | HB1 HG1  0 <-- from peptide bond
ATOM HA      HB  0.10 !      | | | //
GROUP        ! HA-CA--CB--CG-CD
ATOM C        CC 0.34 !      | | | \
ATOM OT1     OC -0.67 !      | HB2 HG2  \ <-- from peptide bond
ATOM OT2     OC -0.67 ! OT1=C
GROUP        !      | <-- from CTER
ATOM CB      CT2 -0.18 !      OT2(-)
ATOM HB1     HA  0.09
ATOM HB2     HA  0.09
GROUP
ATOM CG      CT2 -0.18
ATOM HG1     HA  0.09
ATOM HG2     HA  0.09
GROUP
ATOM CD      C   0.51
ATOM O       O  -0.51
BOND CB CA CG CB CD CG
BOND HT1 N HT2 N HT3 N CA N
BOND CD +N CA HA CB HB1 CB HB2 CG HG1 CG HG2 CA C
BOND C OT2
DOUBLE CD O C OT1
IMPR CD CG +N O !from GLU
IMPR C CA OT1 OT2 !from CTER
DONOR HT1 N !from NTER
DONOR HT2 N !from NTER
DONOR HT3 N !from NTER
ACCEPTOR O CD !from GLU
ACCEPTOR OT1 C !from CTER
ACCEPTOR OT2 C !from CTER
IC N C *CA CB 1.4512 107.2700 121.9000 111.7100 1.5516 !from GLU
IC N C *CA HA 1.4512 107.2700 -118.0600 107.2600 1.0828 !from GLU
IC N CA CB CG 1.4512 111.0400 180.0000 115.6900 1.5557 !from GLU
IC CG CA *CB HB1 1.5557 115.6900 121.2200 108.1600 1.1145 !from GLU
IC CG CA *CB HB2 1.5557 115.6900 -123.6500 109.8100 1.1131 !from GLU
IC CA CB CG CD 1.5516 115.6900 180.0000 115.7300 1.5307 !from GLU
IC CD CB *CG HG1 1.5307 115.7300 117.3800 109.5000 1.1053 !from GLU
IC CD CB *CG HG2 1.5307 115.7300 -121.9600 111.0000 1.1081 !from GLU
IC CB CG CD O 1.5557 115.7300 180.0000 114.9900 1.2590 !from GLU
IC HT1 N CA C 0.0000 0.0000 180.0000 0.0000 0.0000 !from NTER
IC HT2 CA *N HT1 0.0000 0.0000 120.0000 0.0000 0.0000 !from NTER
IC HT3 CA *N HT2 0.0000 0.0000 120.0000 0.0000 0.0000 !from NTER
IC N C OT2 0.0000 0.0000 180.0000 0.0000 0.0000 !from CTER
IC OT2 CA *C OT1 0.0000 0.0000 180.0000 0.0000 0.0000 !from CTER

```

The topology file is finished, you think! Time to run, you say! Wrong. Although we have created the appropriate atomic topology for  $\gamma$ -Glu, we haven't considered the peptide bond between it and the next residue (Cys in the case of GSH). If we were to attempt to create a psf file at this point, psfgen would try to link the residues via a normal peptide bond and generate errors. In order to properly account for the "side chain" peptide bond, we use a patch.



**Patches.** Patches are used to modify the structure. For example, the DISU patch alters two cysteine residues so that there is a disulfide bond between them.

In creating a patch to accurately represent the bond between GGL and CYS, we need to consider how both topologies will be affected.

**14** Find the CYS topology entry in `top_all127_prot_lipid_orn.inp`.

There are two important things to note:

```
IMPR N -C CA HN
IC -C CA *N HN 1.3479 123.9300 180.0000 114.7700 0.9982
```

This is inappropriate for your new GGL residue. Remember that **C** and **CA** are no longer right next to the peptide bond. We need the patch to correct this. For our patch, we need to **DELETE** any inappropriate parameters and build new ones.

Your patch should look something like this:

```
Glutathione Patch

PRES GLNK      0.00 ! linkage for IMAGES or for joining segments
                  ! 1 refers to GGLU (N terminal)
                  ! 2 refers to next (C terminal)
                  ! use in a patch statement
DELETE IMPR 2N 1C 2CA 2HN !Improper specified by IMPR N -C CA HN
DELETE IC 1C 2CA *2N 2HN!Specified by IC -C CA *N HN

IMPR 2N 1CD 2CA 2HN !New improper
IC 1CD 2CA *2N 2HN 1.3479 123.9300 180.0000 114.7700 0.9982 !new IC
```

The last **IC** is needed for building the peptide hydrogen at the GGL-CYS link.

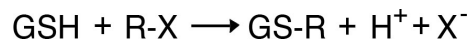
**15** Use your text editor to add the above patch by typing each line into the topology file you just edited.

**16** Click File → Save As..., enter `top_all127_prot_lipid_gsh.inp` as your new filename, and save the file. You have now created a topology file with  $\gamma$ -Glu (and ornithine and D-alanine) included.

**Windows Users:** Make sure you save the file in `.txt` format.

## 2.2 Using the Topology File

Now you are ready to use your new topology file! We will use it to simulate a protein that binds glutathione: the human Pi-class Glutathione S-transferase (GST). GSTs are detoxifying enzymes that conjugate xenobiotic compounds with electrophilic centres to glutathione via the sulfur atom:



In the above equation, **R** is the xenobiotic compound, **X** is the electrophilic centre, and **GS-R** is the glutathione-xenobiotic conjugate.

The addition of glutathione to a xenobiotic increases its solubility and acts as a marker to indicate that the compound is to be excreted by the cell. After excretion, the compound is degraded by the mercapturic acid pathway and excreted by the kidneys. While normal functioning GSTs protect us against toxins in our food and environment, GSTs also attack drugs, and are implicated in cellular resistance to chemotherapy. The human Pi-class GST is a homodimer with 209 residues in each monomer. The monomers have two domains: an N-terminal domain which adopts the thioredoxin fold, found in many GSH-binding proteins, and an all-helical C-terminal domain unique to GSTs. One GSH molecule binds in each active site of GST.

1 Launch VMD by:

- **Unix/Mac OS X Users:** typing `vmd` in a Terminal window.
- **Windows Users:** using the Start menu.

2 In the VMD Main window, click Extensions → Tk Console. Load the pdb structure for glutathione-bound GST, by typing in the VMD TkCon window,

```
mol new 6GSS.pdb
```

Note: Windows users must make sure they are in the 2-glutathione directory. To change to it, type:

```
cd <path to topology-tutorial-files directory>
```

Since the molecule is a homodimer, we must create a separate pdb for each monomer. The monomers exist as separate chains in the pdb.

3 Create pdb files for each GST monomer without ligands. Type the following commands into the TK Console window of VMD:

```
set gstA [atomselect top "chain A and not resname MES GTT"]
$gstA writepdb gst-a.pdb
set gstB [atomselect top "chain B and not resname MES GTT"]
$gstB writepdb gst-b.pdb
```

These commands will write pdb's of the protein without the residues named MES (a buffer molecule from the crystallization mixture) and GTT (glutathione).

4 Create pdb's for the two glutathione peptides and the crystal waters. Type the following commands into the TK Console window of VMD:

```

set gshA [atomselect top "chain A and resname GTT"]
$gshA writepdb gsh-a.pdb
set gshB [atomselect top "chain B and resname GTT"]
$gshB writepdb gsh-b.pdb
set w [atomselect top "resname HOH"]
$w writepdb water.pdb

```

- 5 Edit the files `gsh-a.pdb` and `gsh-b.pdb` with your text editor. Convert the atom names, residue names, and sequence numbers for the glutathione chains so that they are consistent with CHARMM22 and the new GGL residue that you have constructed. It might be helpful to load the glutathione molecule into VMD and label all atoms so that you know which one is which. Furthermore, these changes could be made in VMD via the Tk Console and identifying atoms by their index, but this would be tedious. We will use the text editor since the molecule is only 3 residues long.

The new pdb file, `gsh-a.pdb` should look as follows. File `gsh-b.pdb` is analogous.

Glutathione PDB											
CRYST1	79.220	90.690	69.170	90.00	98.22	90.00	P 1				1
ATOM	1	N	GGL A	1	14.887	10.883	23.275	1.00	73.31		
ATOM	2	CA	GGL A	1	14.820	10.124	24.553	1.00	73.72		
ATOM	3	C	GGL A	1	15.860	9.001	24.508	1.00	73.51		
ATOM	4	OT1	GGL A	1	16.269	8.500	25.582	1.00	72.23		
ATOM	5	OT2	GGL A	1	16.281	8.656	23.381	1.00	73.55		
ATOM	6	CB	GGL A	1	13.422	9.543	24.740	1.00	74.00		
ATOM	7	CG	GGL A	1	13.237	8.843	26.061	1.00	74.50		
ATOM	8	CD	GGL A	1	11.904	8.151	26.176	1.00	75.17		
ATOM	9	O	GGL A	1	11.170	8.008	25.193	1.00	75.35		
ATOM	10	N	CYS A	2	11.589	7.729	27.397	1.00	75.35		
ATOM	11	CA	CYS A	2	10.345	7.036	27.698	1.00	75.19		
ATOM	12	C	CYS A	2	9.835	7.523	29.054	1.00	74.65		
ATOM	13	O	CYS A	2	10.623	7.766	29.973	1.00	73.83		
ATOM	14	CB	CYS A	2	10.585	5.521	27.753	1.00	75.74		
ATOM	15	SG	CYS A	2	11.440	4.813	26.310	1.00	76.23		
ATOM	16	N	GLY A	3	8.520	7.663	29.172	1.00	74.50		
ATOM	17	CA	GLY A	3	7.932	8.116	30.417	1.00	74.66		
ATOM	18	C	GLY A	3	7.340	9.505	30.290	1.00	74.96		
ATOM	19	OT1	GLY A	3	6.438	9.842	31.087	1.00	75.16		
ATOM	20	OT2	GLY A	3	7.761	10.251	29.379	1.00	74.82		
END											

Now, to create a psf of the entire protein with glutathione chains bound, you will need to use the new topology file you created which has `GGL` and `GLNK` topology entries. The psfgen input file `gen-gst.pgn` has been provided for this. You should look at the file before using it. Notice how the GSH segment is handled:

```

segment GSHA {
first none
pdb gsh-a.pdb
}
patch GLNK GSHA:1 GSHA:2

```

The command `first none` means that we apply no N-terminal or C-terminal generating patches to the GGL residue. The `patch` command is the means by which we apply our patch. It is followed by the name of the patch from the topology file (GLNK) and a list of residues and their segment names [segment:residue] to which the patch should be applied (GSHA:1 GSHA:2).



**File gen-gst.pgn.** Other things to note in `gen-gst.pgn` are the aliasing of atoms in the `pdb` to their CHARMM22 equivalents. Take note that the `pdb` designation of the CD1 and CD2 atoms of leucine are reversed in CHARMM22. In this example all HIS residues are aliased to HSE. Strictly speaking, histidine residues should be aliased to HSD, HSE or HSP based on local chemical considerations (hydrogen bonds etc).

- 6 Use `psfgen` to create a `psf` for the system. In the VMD TkCon window, type:

```
source gen-gst.pgn
```

Check the output to make sure no obvious errors occurred. If successful, you will have created files `gst.pdb` and `gst.psf`.



**Psfgen Warnings.** Be sure to look at the messages produced by `psfgen`, as they could indicate a problem with the topology file.

- 7 Check the files visually to make sure they were created properly. In VMD, first delete molecule 6GSS.pdb. Then, load the files `gst.psf` and `gst.pdb` into VMD by typing in the VMD TkCon window:

```
mol new gst.psf
mol addfile gst.pdb
```


Have the hydrogen atoms on GSHA and GSHB been built properly? Unusual geometry could indicate a problem with your topology file.

- 8 Copy the parameter file to be used in your simulation to the current directory. (Alternatively, you could alter the NAMD configuration file to point to the original location of the parameter file.)

- **Unix/Mac OS X Users:** In a terminal window, type:  

```
cp ../1-examine/par_all27_prot_lipid.inp .
```
- **Windows Users:** Use Windows Explorer to copy the file `par_all27_prot_lipid.inp` in the 1-examine directory to the 2-glutathione directory.

At this point, you are ready to run a simulation of your system in vacuum. If you have limited computational power (i.e. those performing this tutorial on a laptop), you should perform a vacuum run at this point and skip Section 2.3. If you have access to greater computational power (i.e. a small cluster), you may skip the vacuum run and continue on to Section 2.3.



**Running.** Run your vacuum simulation.

**Unix/Mac OS X Users:** Type in a terminal window:

- `namd2 gst_vac_mineq.conf > gst_vac_mineq.log &`

**Windows Users:** At the DOS command prompt, you must change to the 2-glutathione directory by typing its full path, and then run NAMD by typing the full path to NAMD, depending on where you installed it. We assume the directory C:\NAMD\.

- `cd <topology-tutorial-files directory>/2-glutathione`
- `C:\NAMD\namd2 gst_vac_mineq.conf > gst_vac_mineq.log`

## 2.3 Solvating and Ionizing


- 1 Place the protein in a water box. In VMD, use the solvate package by typing in the TkCon window:

```
package require solvate
solvate gst.psf gst.pdb -t 5 -o gst_solv
```

- 2 Add ions to model physiological conditions by using VMD's autoionize package. In the TkCon window, type:

```
package require autoionize
autoionize -psf gst_solv.psf -pdb gst_solv.pdb -is 0.2 -o gst_solv_ion
```

- 3 Now lets do some minimization and equilibration using Langevin dynamics. The NAMD configuration file `gst_wb_mineq.conf` has been provided for this. The cell basis vectors should match those output by solvate. The PMEGridSize dimensions should also be appropriate. You should look at the configuration file before running to make sure you understand what it will do.



**Running.** Run your simulation.

**Unix/Mac OS X Users:** Type in a terminal window:

- `namd2 gst_wb_mineq.conf > gst_wb_mineq.log &`

**Windows Users:** At the DOS command prompt, you must change to the 2-glutathione directory by typing its full path, and then run NAMD by typing the full path to NAMD, depending on where you installed it. We assume the directory C:\NAMD\.

- `cd <topology-tutorial-files directory>/2-glutathione`
- `C:\NAMD\namd2 gst_wb_mineq.conf > gst_wb_mineq.log`

As frames are added to the output trajectory (dcd) files, you can load them into VMD as a preliminary check that all is well with the topology file you

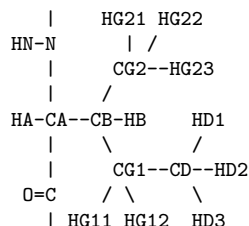


created. Does the GSH molecule behave, or does the geometry distort? Unusual behaviour could indicate a problem in the topology file. If all appears well, you have succeeded!

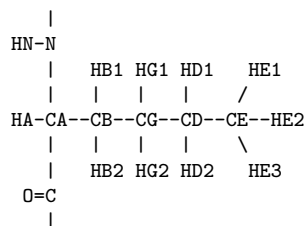
### 3 Solution to Norleucine Problem

Norleucine has the same number of carbon atoms as isoleucine. Both molecules are shown below.

Isoleucine:



Norleucine:



Since norleucine and isoleucine have the same number of carbon atoms, it naturally suggests a source of the topology file for norleucine. Moreover, the norleucine "single chain" structure also reminds us that of lysine.

We've seen the structure and topology file for lysine above, so let's now look at the topology file for isoleucine on the next page.

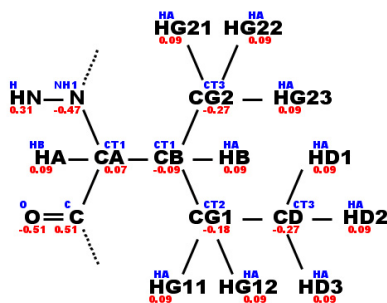


Figure 7: Diagram showing an outline of the topology information for isoleucine. Atom names are displayed in black, atom types in blue, and atomic partial charges in red.

Isoleucine Topology Entry									
RESI	ILE		0.00						
GROUP									
ATOM	N	NH1	-0.47	!		HG21	HG22		
ATOM	HN	H	0.31	!	HN-N		/		
ATOM	CA	CT1	0.07	!		CG2--	HG23		
ATOM	HA	HB	0.09	!		/			
GROUP				!	HA-CA--CB-HB		HD1		
ATOM	CB	CT1	-0.09	!		\	/		
ATOM	HB	HA	0.09	!		CG1--CD--	HD2		
GROUP				!	O=C	/ \	\		
ATOM	CG2	CT3	-0.27	!		HG11	HG12	HD3	
ATOM	HG21	HA	0.09						
ATOM	HG22	HA	0.09						
ATOM	HG23	HA	0.09						
GROUP									
ATOM	CG1	CT2	-0.18						
ATOM	HG11	HA	0.09						
ATOM	HG12	HA	0.09						
GROUP									
ATOM	CD	CT3	-0.27						
ATOM	HD1	HA	0.09						
ATOM	HD2	HA	0.09						
ATOM	HD3	HA	0.09						
GROUP									
ATOM	C	C	0.51						
ATOM	O	O	-0.51						
BOND	CB	CA	CG1	CB	CG2	CB	CD	CG1	
BOND	N	HN	N	CA	C	CA	C	+N	
BOND	CA	HA	CB	HB	CG1	HG11	CG1	HG12	CG2
BOND	CG2	HG22	CG2	HG23	CD	HD1	CD	HD2	CD
DOUBLE	O	C							
IMPR	N	-C	CA	HN	C	CA	+N	O	
DONOR	HN	N							
ACCEPTOR	O	C							
IC	-C	CA	*N	HN	1.3470	124.1600	180.0000	114.1900	0.9978
IC	-C	N	CA	C	1.3470	124.1600	180.0000	106.3500	1.5190
IC	N	CA	C	+N	1.4542	106.3500	180.0000	117.9700	1.3465
IC	+N	CA	*C	O	1.3465	117.9700	180.0000	120.5900	1.2300
IC	CA	C	+N	+CA	1.5190	117.9700	180.0000	124.2100	1.4467
IC	N	C	*CA	CB	1.4542	106.3500	124.2200	112.9300	1.5681
IC	N	C	*CA	HA	1.4542	106.3500	-115.6300	106.8100	1.0826
IC	N	CA	CB	CG1	1.4542	112.7900	180.0000	113.6300	1.5498
IC	CG1	CA	*CB	HB	1.5498	113.6300	114.5500	104.4800	1.1195
IC	CG1	CA	*CB	CG2	1.5498	113.6300	-130.0400	113.9300	1.5452
IC	CA	CB	CG2	HG21	1.5681	113.9300	-171.3000	110.6100	1.1100
IC	HG21	CB	*CG2	HG22	1.1100	110.6100	119.3500	110.9000	1.1102
IC	HG21	CB	*CG2	HG23	1.1100	110.6100	-120.0900	110.9700	1.1105
IC	CA	CB	CG1	CD	1.5681	113.6300	180.0000	114.0900	1.5381
IC	CD	CB	*CG1	HG11	1.5381	114.0900	122.3600	109.7800	1.1130
IC	CD	CB	*CG1	HG12	1.5381	114.0900	-120.5900	108.8900	1.1141
IC	CB	CG1	CD	HD1	1.5498	114.0900	-176.7800	110.3100	1.1115
IC	HD1	CG1	*CD	HD2	1.1115	110.3100	119.7500	110.6500	1.1113
IC	HD1	CG1	*CD	HD3	1.1115	110.3100	-119.7000	111.0200	1.1103

First of all, we will split norleucine into several atom groups, each with an integer charge, and each of which resembles similar atom groups in either lysine or isoleucine. As shown in figure 8, the backbone group and the -CH2 groups are very similar to those of lysine, so we will use the corresponding atom groups in lysine to assign atom types and charges for norleucine.

The terminal methyl group, -CH3, in norleucine is very similar to that of isoleucine. We will use those isoleucine group properties for the norleucine

-CH<sub>3</sub>.

The figure below shows how we have grouped the norleucine molecule and how we will piece together its topology from existing topologies.

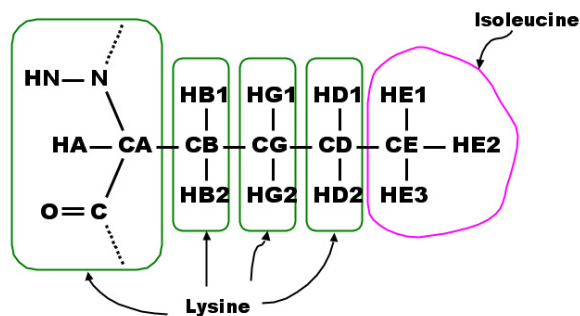


Figure 8: Diagram showing the structure of norleucine. Groups circled in green have corresponding groups in the lysine topology entry, while the terminal methyl group circled in pink is similar to that of isoleucine.

After piecing the molecule together from existing groups, we can create a topology entry for norleucine. It should look like the one on the next page.

```

Norleucine Topology Entry

RESI NLE          0.00
!!! Lysine groups:
GROUP
ATOM N    NH1    -0.47 !      |
ATOM HN   H      0.31 !    HN-N
ATOM CA   CT1    0.07 !      | HB1 HG1 HD1    HE1
ATOM HA   HB     0.09 !      | | | | /
GROUP     !    HA-CA--CB--CG--CD--CE--HE2
ATOM CB   CT2    -0.18 !      | | | | \
ATOM HB1  HA     0.09 !      | HB2 HG2 HD2    HE3
ATOM HB2  HA     0.09 !    O=C
GROUP     !      |
ATOM CG   CT2    -0.18
ATOM HG1  HA     0.09
ATOM HG2  HA     0.09
GROUP
ATOM CD   CT2    -0.18
ATOM HD1  HA     0.09
ATOM HD2  HA     0.09
GROUP
ATOM C    C       0.51
ATOM O    O      -0.51
!!! Isoleucine groups:
GROUP
ATOM CE   CT3    -0.27
ATOM HE1  HA     0.09
ATOM HE2  HA     0.09
ATOM HE3  HA     0.09
BOND CB CA    CG CB    CD CG    CE CD
BOND N HN    N CA    C CA
BOND C +N    CA HA    CB HB1    CB HB2    CG HG1
BOND CG HG2    CD HD1    CD HD2
DOUBLE O C
BOND CE HE1    CE HE2    CE HE3
IMPR N -C CA HN    C CA +N O
DONOR HN N
ACCEPTOR O C
!!! Internal coordinate entries
!!! Lysine ICs:
IC -C    CA    *N    HN    1.3482 123.5700 180.0000 115.1100 0.9988
IC -C    N    CA    C    1.3482 123.5700 180.0000 107.2900 1.5187
IC N    CA    C    +N    1.4504 107.2900 180.0000 117.2700 1.3478
IC +N    CA    *C    O    1.3478 117.2700 180.0000 120.7900 1.2277
IC CA    C    +N    +CA    1.5187 117.2700 180.0000 124.9100 1.4487
IC N    C    *CA    CB    1.4504 107.2900 122.2300 111.3600 1.5568
IC N    C    *CA    HA    1.4504 107.2900 -116.8800 107.3600 1.0833
IC N    CA    CB    CG    1.4504 111.4700 180.0000 115.7600 1.5435
IC CG    CA    *CB    HB1    1.5435 115.7600 120.9000 107.1100 1.1146
IC CG    CA    *CB    HB2    1.5435 115.7600 -124.4800 108.9900 1.1131
IC CA    CB    CG    CD    1.5568 115.7600 180.0000 113.2800 1.5397
IC CD    CB    *CG    HG1    1.5397 113.2800 120.7400 109.1000 1.1138
IC CD    CB    *CG    HG2    1.5397 113.2800 -122.3400 108.9900 1.1143
IC CB    CG    CD    CE    1.5435 113.2800 180.0000 112.3300 1.5350
!!! Isoleucine ICs:
IC CE    CG    *CD    HD1    1.5381 114.0900 122.3600 109.7800 1.1130
IC CE    CG    *CD    HD2    1.5381 114.0900 -120.5900 108.8900 1.1141
IC CG    CD    CE    HE1    1.5498 114.0900 -176.7800 110.3100 1.1115
IC HE1    CD    *CE    HE2    1.1115 110.3100 119.7500 110.6500 1.1113
IC HE1    CD    *CE    HE3    1.1115 110.3100 -119.7000 111.0200 1.1103

```

Recall that the ! sign is simply for comments, and helps to make the topology file more readable and user-friendly.

Note that since we have pieced together the norleucine molecule from lysine

and isoleucine, the IC records of norleucine also come in two parts: lysine-similar and isoleucine-similar. For the N-terminal half of norleucine, from carbon atom CA to CG, we use ICs lysine's topology file, as shown. As for C-terminal half of norleucine, from the carbon atom CE to CG, we can refer to the corresponding atoms CG1 and CD of isoleucine to get the IC records.

Before simulating a protein with norleucine, you should create the above topology entry in your topology file, and rename the file appropriately, as done with D-alanine. The file `top_all127_prot_lipid_nle.inp` has been provided for you in the directory `3-solution/`.

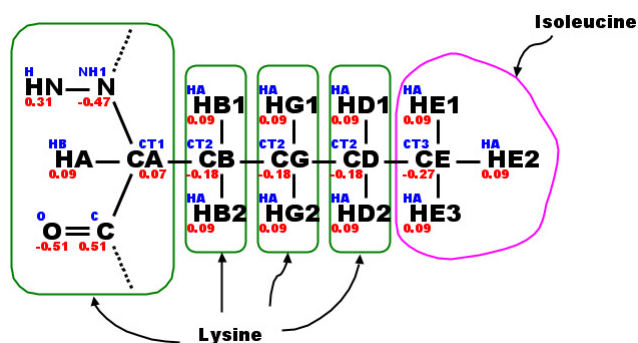


Figure 9: Diagram showing a complete outline of the topology information for norleucine. The molecule is pieced together using groups from lysine and isoleucine as shown. The atoms types and charges, which have been taken from corresponding parts in lysine and isoleucine are shown. Atom names are displayed in black, atom types in blue, and atomic partial charges in red.

We have created a topology entry for norleucine by splitting it into two components, each of which is similar to an amino acid with known topology file. Now let's have a look at a specific protein that contains norleucine, HIV-1 GAG(Group-specific Antigen) protein, which is a structural protein in HIV. The pdb file `1FGL.pdb` contains a structure of HIV-1 GAG. Chain B is part of HIV-1 GAG which contains norleucine.

Load the pdb file into VMD and select the Coloring Method to be "ResType". All the residues are now colored by white, green, red and blue, which correspond to nonpolar, polar, acidic and basic residues. The only exception is norleucine, which is colored cyan, because VMD doesn't have information about its residue type. It is "unassigned".

With the topology file we just generated, we can build a psf file for the whole protein. First, load `1FGL.pdb` into VMD and write a separate pdb file for each chain:

```

set chaina [atomselect top "chain A"]
$chaina writepdb 1FGL-chainA.pdb
set chainb [atomselect top "chain B"]
$chainb writepdb 1FGL-chainB.pdb
set water [atomselect top "water"]
$water writepdb 1FGL-water.pdb

```

The file `nle.pgn` is the `psfgen` script for building the `psf` and `pdb` file for the whole protein. Note that `top_all122_prot_lipid_nle.inp` is the topology file specified in `nle.pgn`. It is simply the file `top_all122_prot_lipid_gsh.inp` created earlier, but with our norleucine topology entry added. You can open it and browse the file by typing `nedit top_all122_prot_lipid_nle.inp` in a terminal window.

Now, generate the `psf` file for HIV-1 GAG by going to the VMD TkCon window and typing:

```
source nle.pgn
```

You will generate files `nle.psf` and `nle.pdb`, which you can use to run a NAMD simulation!

## Acknowledgments

Development of this tutorial was supported by the National Institutes of Health (P41-RR005969 - Resource for Macromolecular Modeling and Bioinformatics).